This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1.   (Previously Presented) A method for upgrading managed state for a JAVA based application, comprising:

executing a JAVA module on a server, wherein the JAVA module is in a middle-tier between a client browser and databases, the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean;

generating an upgraded state object, the JAVA module including the upgraded state object, wherein the upgraded state object is generated by upgrading a physical schema using data stored in a repository that is part of the databases;

transferring the state stored in the original state object to the upgraded state object, wherein the original state object is upgraded in the JAVA module;

generating an upgraded entity bean using data stored in the system repository; and

providing state management for the original entity bean using the upgraded state object.

2.   (Cancelled)

3.   (Original) A method as recited in claim 1, further comprising the operation of managing the state of the upgraded entity bean using the upgraded state object.

4.   (Original) A method as recited in claim 3, wherein both the original entity bean and the original state object are disabled.

5.   (Cancelled)

6.   (Previously Presented) A method as recited in claim 4 wherein functionality of the JAVA module is not disrupted when the upgraded state object is generated.

7.   (Previously Presented) A method as recited in claim 4, wherein functionality of the JAVA application is not disrupted when the JAVA module is upgraded.

8.   (Previously Presented) A JAVA platform capable of performing an online upgrade on a JAVA application, the JAVA platform comprising:

a JAVA module in a middle tier between a client browser and databases, the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, wherein the original state object storing a state of the original entity bean, and wherein the state object provides state management for the original entity bean; and

a repository that is part of the databases and having upgraded class files for the original entity bean and upgraded class files for the original state object,

wherein the original state object is upgraded by generating an upgraded state object, the JAVA module including the upgraded state object, using upgraded class files from the repository, and transferring the state stored in the original state object to the upgraded state object wherein the original state object is upgraded in the JAVA module; and

an upgrade entity bean is created using data from the repository as the JAVA platform is upgraded.

9.   (Cancelled)

10.   (Previously Presented) A JAVA platform as recited in claim 8, wherein the state of the upgraded entity bean is managed using the upgraded state object.

11.   (Previously Presented) A JAVA platform as recited in claim 10, wherein both the original entity bean and the original state object are disabled.

12.   (Previously Presented) A JAVA platform as recited in claim 8, wherein the upgraded state object is generated by upgrading a physical schema using data stored in the repository.

13.   (Previously Presented) A JAVA platform as recited in claim 12, wherein functionality of the JAVA module is not disrupted when the upgraded state object is generated.

14.   (Previously Presented) A JAVA platform as recited in claim 13, wherein functionality of the JAVA application is not disrupted when the JAVA module is upgraded.

15.   (Previously Presented) A method for upgrading a JAVA application having a managed application state, comprising the operations of:

executing a JAVA module on a server, wherein the JAVA module is in a middle tier between a client browser and databases, the JAVA module includes at least one original entity bean and at least one original state object in communication with the original entity bean, the original state object storing a state of the original entity bean;

generating an upgraded state object, the JAVA module including the upgraded state object, using data stored in a system repository that is part of the databases;

transferring the state stored in the original state object to the upgraded state object, wherein the original state object is upgraded in the JAVA module;

providing state management for the original entity bean using the upgraded state object;

generating an upgraded entity bean using data stored in the system repository;

providing state management for the upgraded entity bean using the upgraded state object; and

disabling both the original entity bean and the original state object.

16.   (Previously Presented) A method as recited in claim 15, wherein the upgraded state object is generated by upgrading a physical schema using data stored in the repository.

17.   (Previously Presented) A method as recited in claim 16, wherein functionality of the

JAVA module is not disrupted when the upgraded state object is generated.

18.   (Previously Presented) A method as recited in claim 17, wherein functionality of the

JAVA application is not disrupted when the JAVA module is upgraded.

19.   (Previously Presented) A method as recited in claim 18, wherein the original state

object and the upgraded state object are respectively classified into a particular state

management unit.

20.   (Original) A method as recited in claim 19, wherein the particular state management

unit is used to facilitate upgrading of the original state object.